03/23/05

AF
EJW

CERTIFICATE OF MAILING via EXPRESS MAIL
37 C.F.R. § 1.10

PURSUANT TO 37 C.F.R. § 1.10, I HEREBY CERTIFY THAT I HAVE A REASONABLE BASIS FOR BELIEF THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE AS EXPRESS MAIL POST OFFICE TO ADDRESSEE, ON THE DATE BELOW, IN AN ENVELOPE ADDRESSED TO:

MAIL STOP APPEAL BRIEF-PATENTS
HONORABLE COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

Ronald L Chichester

RONALD L. CHICHESTER REG. NO. 36,765

MARCH 21, 2005 **Ev590022408US**
DATE **EXPRESS MAIL LABEL**

APPLICATION NUMBER:	09/892,837
FILING DATE:	JUNE 27, 2001
FIRST NAMED INVENTOR:	KLINDT, ET AL.
GROUP ART UNIT:	2172
EXAMINER:	CAM Y T TRUONG
TITLE:	"COPYING A PORTION OF A DATABASE STRUCTURE ASSOCIATED WITH A QUERY"
ATTORNEY DOCKET NUMBER:	069092.0118

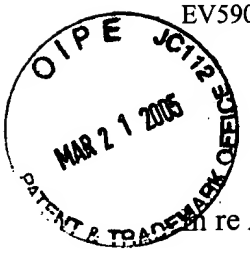
INCLUDED IN THIS MAILING FOR THE ABOVE-REFERENCED PATENT APPLICATION ARE:

1. APPELLANT'S NEW BRIEF IN RESPONSE TO NOTICE OF NON-COMPLIANCE UNDER 37 C.F.R. § 1.192 (C) (IN TRIPLICATE); AND
2. RETURN POSTCARD TO ACKNOWLEDGE RECEIPT OF ABOVE ITEMS.

ATTORNEY CONTACT: RONALD L. CHICHESTER REG. NO. 36,765
BAKER BOTTS L.L.P. PTO ID# 023640
TELEPHONE: 713.229.1341
FACSIMILE: 713.229.7741
EMAIL: RONALD.CHICHESTER@BAKERBOTTS.COM
PTO CUSTOMER NUMBER:

023640

EV590022408US



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:		§	
	KLINDT, ET AL.	§	Group Art Unit: **2172**
		§	
Serial No.:	**09/892,837**	§	
		§	Examiner: **TRUONG, CAM Y.T.**
Filed:	**06/27/2001**	§	
		§	
Title:	**"COPYING A PORTION OF A DATABASE STRUCTURE ASSOCIATED WITH A QUERY"**	§	Atty. Docket No.: **069092.0118**

Mail Stop Appeal Brief - Patents
Honorable Commissioner for Patents
Washington, D.C. 20231

Attention: Board of Patent Appeals and Interferences

Dear Sir:

APPELLANTS' NEW BRIEF (37 C.F.R. § 1.192)

This brief is submitted in support of Applicants' Notice of Appeal from the decision dated January 9, 2004 of the Examiner finally rejecting claims 1 through 30 of the subject application. This Appellant's Brief is in response the Notice of Non-Compliance that was mailed on February 19, 2005. The one-month shortened statutory period for filing the new Appeal Brief in response to the Notice of Non-Compliance was Saturday, March 19, 2005, and this new Appeal Brief is being filed on the next business day, Monday, March 21, 2005 and is therefore considered timely filed.

This new brief is transmitted in triplicate per 37 C.F.R. § 1.192.

I. IDENTIFICATION OF THE REAL PARTY OF INTEREST

The real party in interest is:

NCR Corporation
1700 S. Patterson Blvd.
Dayton, OH 45479

by virtue of assignments by the inventors Jerry L. Klindt and Paul L. Sinclair.

II. IDENTIFICATION OF RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF ALL THE CLAIMS, PENDING OR CANCELLED, AND IDENTIFYING THE CLAIMS APPEALED

The application as originally filed contained 30 claims. All pending claims of the present application are reproduced in Appendix A, attached hereto.

IV. STATUS OF ANY AMENDMENT FILED SUBSEQUENT TO FINAL REJECTION

A response to the final office was filed by the Applicants on April 9, 2004. An Advisory Action was mailed by the examiner and was received by the Applicants on May 6, 2004. The Advisory Action indicated that the proposed amendments made in the response to the final office action would *not* be entered. Consequently, the status of the claims is as recited in Appendix A.

V. SUMMARY OF THE INVENTION

The present invention concerns databases, particularly relational databases that can perform multiple functions through use of appropriately formatted queries. The queries can be in the form of a structured query language ("SQL") query. Page 2, lines 14-15. Relational databases can contain one or more databases, and each database can contain one or more tables

of information. Page 2, lines 13-24. Software objects can be mapped to relational database tables using mapping techniques. Typically, the mapping technique employs an object definition that is kept within the database is used to generate the object from data stored in the database. More specifically, the present invention enables the copying of a portion of a database structure that contains one or more objects. Page 3, lines 9-11; Figure 6, element 620; Page 9, lines 4-15. The method of the present invention receives a query, such as an SQL query, and recursively retrieves the object definitions for the queried objects. Page 3, lines 13-14; Page 5, lines 10-19. The ultimate result is an ordered set of object definitions that were associated with the query. Page 3, lines 9-11. The ordered set of object definitions can be used to build a copy of a portion of the database structure on the same or a different computer. Page 3, lines 28-29; Page 9, lines 22-23. For example, the database structure can be used to create a second database (with different data) or the database structure could be populated with data from the original database, thereby making a copy of a portion of the original database. Page 3, line 28 through Page 4, line 2; Figure 6, element 620; Page 9, lines 4-23.

The step of recursively retrieving object definitions can include a sub-step of recursively identifying objects that are associated with the query and categorizing them into a category. Page 3, lines 14-15; page 5, line 10-12. The step of recursively retrieving can be augmented by and retrieving the object definition for the categorized objects using a tool that corresponds to the category to which the object has been categorized. Page 3, lines 15-17; Page 9, lines 10-24; Page 11, lines 4-7. The categories can include tables, views, join indexes, triggers and macros. Page 3, line 17; Page 11, lines 4-7. The tool may be, for example, a SHOW statement that is correlated to the category (i.e., SHOW TABLE when the cateogry is table, SHOW VIEW wne the category is view, and so on). Page 3, lines 17-22; Page 11, lines 4-7.

The method may further include the step of changing the ordered set of object definitions. Page 3, line 28; Figure 6, element 605; Page 9. lines 1-3. The step of changing may include the reordering of the object definitions such that base objects are placed higher in the order so that those base objects exist when they are referenced by other objects. Page 3, line 29 through Page 4, line 9. Page 9, lines 4-15. The method may further include the step of looking for references to the database objects in a data dictionary while recursively retrieving the object definitions. Page 4, lines 20-24; Page 9, line 24 through Page 10, line 4.

VI. CONCISE STATEMENT OF THE ISSUES PRESENTED FOR REVIEW

Claims 1, 2, 9-12, 19-22, 29-30 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Bapat (U.S. Patent No. 5,295,256).

With respect to claims 1, 11 and 21, the examiner argued that Bapat teaches the claimed limitations: "recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of objects definitions." Applicants do not believe that Bapat discloses, teaches or suggests the limitations of claims 1, 11 and 21, and by extension, all claims dependent upon independent claims 1, 11 and 21.

VII. GROUPING OF CLAIMS

All pending claims stand or fall together.

VIII. ARGUMENTS OF THE APPELLANTS, WITH EACH ISSUE IN SEPARATE HEADINGS, WITH RESPECT TO EACH ISSUE PRESENTED FOR REVIEW

A. Bapat does not teach or suggest copying a portion of a portion of a database structure by recursively retrieving object definitions in response to a query as claimed in claims 1, 11, and 21.

The Examiner states in her response to Arguments (page 2, Final Office Action, Paper

No. 7) made in the previous Office Action Response that:

> Applicant's arguments filed 10/29/03 have been fully considered but they are not persuasive. Claims 1-30 are pending in this Office Action.
>
> Applicant argues that Bapat does not teach "each object in the hierarchy is associated with a query and recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definition." Bapat teaches that each object in the hierarchy is processed by a loop beginning at step 292, which selects every class definition in the object class hierarchy. Each class definition is retrieved at step 292 until no more objects in hierarchy are processed. Each object in the hierarchy is processed when the application make a call to the new method such as retrieve. It means that objects in the hierarchy are associated with the application's call. The application's call is represented as a query. Recursively is defined as to repeat again. Since each class definition, which is retrieved in a loop, is repeated many times until no more objects class in hierarchy is processed, thus, each class definition in hierarchy is retrieved recursively and in order. Each class definition contains one or more object definitions. For the above reasons, definitely, when system retrieves each class definition in hierarchy, the system retrieves each object definition of the class definition to produce an ordered set of object definitions too (fig. 13, col. 10, lines 35-65; col. 6, lines 65-67).
>
> Applicant argues that Bapat does not teach "building a copy of the database structure using the ordered set of object definitions." Bapat teaches that a class hierarchy structure is translated into the relational table structure by mapping parent class 22 to a table 32. In a similar manner, class 24 is mapped to a table 38. A class in hierarchy includes objects definitions. Since, the system maps parent class 22 to a table 32 before mapping class 24 to a table 38, thus the system retrieves set of class definitions or object definitions in order. By mapping a class hierarchy structure into the relational table, the system creates a relational table structure, which contains all of classes or objects of the hierarchy structure. Thus, the relational table structures are presented as a copy of the hierarchy class structure. The hierarchy class structure is a data structure (col. 6, lines 60-65; col. 7, lines 1-5, figs. 1-3).
>
> Applicant argues that Bapat does not teach "categorizing each identified object into a category." Bapat teaches class site contains object name, address, phone number, site-category. Class vendor contain object vendor name, object vendor address, object vendor phone number. The above information shows that each

object is classified into different classes. Each class is represented as a category (col. 19, lines 35-50).

For the above reason, examiner believed that rejection of the last office action was proper.

Applicants' arguments filed on 10/29/2003 are reproduced below:

Applicants respectfully disagree and submit that Bapat does not disclose, teach, or suggest each and every feature recited in Applicants' independent Claims 1, 11, and 21. For example, Bapat does not disclose teach or suggest "recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definitions," as recited in Applicants' Claim 1. To the contrary, Bapat merely discloses "a translator which converts object-oriented representations of data into relational tables." (Column 5, lines 33-36). For example, Bapat provides that a class construct may include a parent class 22 having "several attributes associated with it," a derived class 24 that "inherits the attributes of the patent class 22 and . . . has its own attributes," and a derived class 26 that "inherits attributes from derived class 24 and parent class 22 and contributes its own attributes." (Column 6, lines 34-49). Bapat provides "a mechanism for mapping this hierarchical schema into a relational table schema." (Column 6, lines 55-57). "[The] parent class 22 is mapped to a table 32 which is named after the class name (P1), . . . a table 38 is constructed representing derived class 24, . . . [and] a third table 44 is constructed to represent derived class 26." (Column 6, line 62 through Column 7, line 9). Accordingly, Bapat is limited to a system for imposing structure on a hierarchical schema to produce tables representing each class in the hierarchical schema. As such, although Bapat discloses that "each object in the hierarchy is processed by a loop beginning at step 292 which selects every class definition in the object class hierarchy," the class definitions are merely used with a "CREATE TABLE" SQL command to construct a table with the same table name. (Column 10, lines 38-45). Thus, Bapat does not teach that each object in the hierarchy is associated with a query as required by Claim 1. For at least these reasons, Applicants respectfully submit that Bapat does not disclose "recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definitions," as recited in Applicants' Claim 1.

Additionally, Applicants submit that Bapat does not disclose, teach, or suggest, "building a copy of the database

structure using the ordered set of object definitions," as recited in Applicants' Claim 1. The Examiner has acknowledged that the recited features and operation are absent from the teachings of Bapat but states that "it would have been obvious to a person of ordinary skill in the art at the time the invention was made to apply Bapat's teaching of restoring each retrieved class or struct definition in each file in class hierarchy table and class attribute table in order to allow any user to create a new data structure in order." (Office Action, page 3). Applicants respectfully disagree. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the Examiner must present a convincing line of reasoning as to why the art is one that would have found the claim to be obvious in light of the teachings of the references." M.P.E.P. § 706.02(j) (citing *Ex parte Clapp*, 227 U.S.P.Q. 972, 973 (Bd. Pat. App. & Inter. 1985)). The Examiner presents neither. Not only does the Examiner fail to cite any support for his conclusion, but the Examiner's conclusion does not follow from the disclosure of Bapat. As discussed above, Bapat relates to a system for translating data from a hierarchy scheme to a relational scheme for storage purposes. As such, the system of Bapat uses object definitions for a class to impose structure on a hierarchical schema to produce tables representing each class in the hierarchical schema. If a "proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." M.P.E.P. § 2143.01. Applicants respectfully submit that the modification of Bapat, as suggested by the Examiner, would "change the principle of operation" of Bapat. Accordingly, Bapat also does not disclose, teach, or suggest "building a copy of the database structure using the ordered set of object definitions," as recited in Applicants' independent Claim 1.

Independent Claims 11 and 21 recite certain features and operations that are similar to the features and operations discussed above. For example, Claim 11 recites a computer-readable medium operable to "recursively retrieve object definitions for one or more database objects associated with a query to produce an ordered set of object definitions" and "build a copy of the database structure using the ordered set of object definitions." Claim 21 recites "recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definitions." Thus, for reasons similar to those discussed above with regard to Claim 1, Applicants respectfully submit that Bapat does not disclose, teach, or suggest each and every feature

and operation as set forth in Applicants' Claims 11 and 21.

For these reasons, Applicants respectfully request that the rejections of Claims 1, 11 and 21 be withdrawn.

The Applicants reiterate the above argument with the same force and effect as originally submitted. The Applicants further note that Bapat, upon which the rejection of the independent claims stand or fall, is directed to mapping objects into relational database tables. Bapat does not teach or suggest mapping *some* of the objects, but rather *all* of the objects. In contrast, the present invention obtains information on only a portion of the database. Moreover, Bapat does not teach or suggest a *recursive* retrieval of objects from *within* the database, but rather a systematic generation of relational database representations that are put *into* the Bapat database. Bapat does not teach or suggest recursive retrieval of object definitions from within the database because Bapat is directed to another purpose -- one that is unrelated to retrieving object definitions from within the database. That is why Bapat does not disclose, teach, or suggest the limitations of independent claims 1, 11 and 21, and that is why Bapat does not render independent claims 1, 11 and 21 obvious under 35 U.S.C. §103(a). M.P.E.P. §706.02(j), *Ex parte Clapp, supra.* For these reasons, Applicants respectfully request that the rejections of Claims 1, 11 and 21 be withdrawn.

Claims 2-10 depend from claim 1, claims 12-20 depend from claim 11, and claims 22-30 depend from claim 21, and contain all limitations thereof. Therefore the dependent claims should be allowable for the same reasons as independent claims 1, 11 and 21. Applicants respectfully submit that the cited reference does not disclosed an apparatus, system or method which anticipates the present invention as claimed in independent claims 1, 11 and 21, or in the claims that depend upon claims 1, 11 and 21. Therefore, it is respectfully submitted that all

claims are allowable over the prior art of record, and such allowance is earnestly solicited.

Summary

In light of the foregoing, Applicants respectfully request that the final rejection of the pending claims should be reversed and the application be remanded for allowance of the pending claims, or, alternatively, remand the application for further examination if appropriate references can be found by the examiner.

Applicants have previously submitted the fee for filing this appeal. Applicants believe that there are no additional fees due in association with the filing of this Appeal Brief. However, should the Commissioner deem any additional fees as being due, including any fees for any additional extensions of time, the Commissioner is requested to accept this as a Petition therefore, and is hereby authorized to charge any additional fees due, or to credit any overpayment, to Baker Botts L.L.P. Deposit Account No. **02-0383**, Order Number **069092.0118** under 37 C.F.R. § 1.16 or § 1.17.

Respectfully submitted,

BAKER BOTTS L.L.P. (023640)

By: 

Ronald L. Chichester
Registration No. 36,765
Baker Botts L.L.P.
One Shell Plaza
910 Louisiana Street
Houston, Texas 77002-4995
Telephone: 713.229.1341
Facsimile: 713.229.7741
EMail: Ronald.Chichester@bakerbotts.com
ATTORNEY FOR APPLICANTS

Date: March 21, 2005

APPENDIX A

1. (original) A method for copying a portion of a database structure, the database including one or more database objects, the method comprising

recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definitions;

building a copy of the database structure using the ordered set of object definitions.

2. (original) The method of claim 1, where recursively retrieving object definitions includes

recursively identifying objects associated with the query;

categorizing each identified object into a category;

retrieving an object definition for each identified object using a tool corresponding to the category with which the identified object is connected.

3. (original) The method of claim 2, where the categories include tables, views, join indexes, triggers and macros.

4. (original) The method of claim 2, where the tool is

a. a SHOW VIEW statement if the identified object is categorized as a view;

b. a SHOW TABLE statement if the identified object is categorized as a table;

c. a SHOW JOIN INDEX statement if the identified object is categorized as a join index;

d. a SHOW TRIGGER statement if the identified object is categorized as a trigger;

e. a SHOW MACRO statement if the identified object is categorized as a macro.

5. (original) The method of claim 1, where recursively retrieving object definitions includes

a. retrieving unretrieved object definitions for a set of objects known to be associated with the query;

b. adding to the set of objects known to be associated with the query those objects contained in the retrieved object definitions that are not already in the set of objects known to be associated with the query;

c. repeating items a and b until no new objects are added to the set of objects known to be associated with the query.

6. (original) The method of claim 1, further comprising sending the ordered set of object definitions from a first computer to a second computer.

7. (original) The method of claim 1, further comprising changing the order of the ordered set of object definitions.

8. (original) The method of claim 7, where changing the order of the ordered set of object definitions includes reordering the object definitions so that each object definition is ordered before the definition of any object that references it.

9. (original) The method of claim 1, where the object definitions are ordered so that each object definition is ordered before the definition of any object that references it.

10. (original) The method of claim 1, where recursively retrieving object definitions for one or more database objects includes looking for references to the one or more database objects in a data dictionary.

11. (original) A computer-readable medium containing computer-executable code for instructing a computer to:

recursively retrieve object definitions for one or more database objects associated with a query to produce an ordered set of object definitions;

build a copy of the database structure using the ordered set of object definitions.

12. (original) The computer-executable code of claim 11, in which, when recursively retrieving object definitions, the computer:

recursively identifies objects associated with the query;

categorizes each identified object into a category;

retrieves an object definition for each identified object using a tool corresponding to the category with which the identified object is connected.

13. (original) The computer-executable code of claim 12, where the categories include tables, views, join indexes, triggers and macros.

14. (original) The computer-executable code of claim 12, where the tool is

a. a SHOW VIEW statement if the identified object is categorized as a view;

b. a SHOW TABLE statement if the identified object is categorized as a table;

c. a SHOW JOIN INDEX statement if the identified object is categorized as a join index;

d. a SHOW TRIGGER statement if the identified object is categorized as a trigger;

e. a SHOW MACRO statement if the identified object is categorized as a macro.

15. (original) The computer-executable code of claim 11, in which, when recursively retrieving object definitions, the computer:

a. retrieves unretrieved object definitions for a set of objects known to be associated with the query;

b. adds to the set of objects known to be associated with the query those objects contained in the retrieved object definitions that are not already in the set of objects known to be associated with the query;

c. repeats items a and b until no new objects are added to the set of objects known to be associated with the query.

16. (original) The computer-executable code of claim 11, further comprising computer-executable code instructing the computer to send the object definitions from a first computer to a second computer.

17. (original) The method of claim 11, further comprising computer-executable code instructing the computer to change the order of the ordered set of object definitions.

18. (original) The computer-executable code of claim 17, in which, when changing the order of the ordered set of object definitions, the computer reorders the object definitions so that each object definition is ordered before the definition of any object that references it.

19. (original) The computer-executable code of claim 11, in which, when storing the definitions, the computer stores the definitions so that each object definition is ordered before the definition of any object that references it.

20. (original) The computer executable code of claim 11, in which, when recursively retrieving object definitions for one or more database objects, the computer looks for references to the one or more database objects in a data dictionary.

21. (original) A package of data useful in building a copy of a database structure generated in accordance with the following act:

recursively retrieving object definitions for one or more database objects associated with a query to produce an ordered set of object definitions.

22. (original) The package of data of claim 21, where the object definitions are recursively retrieved in accordance with the following acts:

recursively identifying objects associated with the query;

categorizing each identified object into a category;

retrieving an object definition for each identified object using a tool corresponding to the category with which the identified object is connected.

23. (original) The package of data of claim 22, where the categories include tables, views, join indexes, triggers and macros.

24. (original) The package of data of claim 22, where the tool is

a. a SHOW VIEW statement if the identified object is categorized as a view;

b. a SHOW TABLE statement if the identified object is categorized as a table;

c. a SHOW JOIN INDEX statement if the identified object is categorized as a join index;

d. a SHOW TRIGGER statement if the identified object is categorized as a trigger;

e. a SHOW MACRO statement if the identified object is categorized as a macro.

25. (original) The package of data of claim 21, where the object definitions are recursively retrieved in accordance with the following acts:

a. retrieving unretrieved object definitions for a set of objects known to be associated with the query;

b. adding to the set of objects known to be associated with the query those objects contained in the retrieved object definitions that are not already in the set of objects known to be associated with the query;

c. repeating items a and b until no new objects are added to the set of objects known to be associated with the query.

26. (original) The package of data of claim 21, further comprising the act of sending the ordered set of object definitions from a first computer to a second computer.

27. (original) The package of data of claim 21, further comprising the act of instructing the computer to change the order of the ordered set of object definitions.

28. (original) The package of data of claim 27, where the act of changing the order of the ordered set of object definitions includes reordering the object definitions so that each object is ordered before the definition of any object that references it.

29. (original) The package of data of claim 21, where the act of storing object definitions includes storing the object definitions so that each object definition is ordered before the definition of any object that references it.

30. (original) The package of data of claim 21, where the act of recursively retrieving object definitions for one or more database objects includes looking for references to the one or more database objects in a data dictionary.